REMARKS

In view of the following remarks responsive to the Final Office Action dated July 24, 2007, Applicant respectfully requests favorable reconsideration of this application.

The Office rejected claims 1-4, 7, 11-28, 31, and 35-48 under 35 U.S.C. 102(e) as being anticipated by Barnett. The Office further rejected claims 5, 6, 8, 9, 29, 30, 32, and 33 under 35 U.S.C. 103(a) as unpatentable over Barnett in view of Sycara. Finally, the Office rejected claims 10 and 34 as unpatentable under 35 U.S.C. 103(a) over Barnett in view of Project JXTA.

These rejections are the same rejections asserted in the previous Office Action. The Office has kindly provided responses to most of Applicants' arguments made against these rejections in response to the previous Office Action. However, those responses are erroneous, focus on insignificant details while failing to appreciate the big picture, and ignore significant claim recitations.

The present invention pertains to a method and apparatus for dynamically reconfiguring web services software modules amongst a plurality of different servers on the web in order to facilitate their use by client machines over the web. Barnett, on the other hand, addresses client use of web services and has nothing to do with the dynamic reconfiguration and maintenance of web services software modules amongst the servers. The present invention has almost nothing to do with the accessing and use of those web services software modules by the actual clients (although, of course, the ultimate goal of the present invention is to facilitate client use of the web services software modules). From the client's perspective, the client's use of web services software modules is conventional.

The Office is trying to fit a square peg in a round hole by applying Barnett to the present invention.

Having said that, however, Applicant does not dispute that one must look at the claim language to determine whether that language is broad enough to unintentionally read on prior art that is very different from the spirit of the invention (such as Barnett),

whether or not it is the Applicant's intent to cover such subject matter.

In this case, however, the claims do not read on Barnett. Barnett does not discuss in any way, shape, or form, reconfiguration and maintenance of web services software modules amongst a plurality of servers. In fact, Figures 1 and 2 of Barnett, which are the only figures that show any of the nodes of a network, show only servers 230 having web services software modules on them. Web server 200 does not have any web services software modules on it. Rather, web server 200 services client 205 with the service of sending data to the client 205 so that the client can locally maintain a list of web services that are available to it through other servers, i.e., look up servers 230.

In short, in Barnett, server 200 tells client browser 205 what web services are available on other servers 230. This is completely different from the present invention in which a server at which web services software modules are locally available tells other servers what web services software modules are available at itself.

The claim language clearly distinguishes over Barnett.

Let us consider independent claim 1 for example.

The First Paragraph of Claim 1

The first claim element comprises instructions for "determining and describing web services software modules that are available at a corresponding, local network node". The Office asserts that this is found in Barnett in paragraph [0038]. However, as noted above, paragraph [0038] discusses LoadBalancer 270 on server 200 and particularly says that LoadBalancer 270 "maintains a list of registries (look up servers not expressly shown in 270 of Fig. 2) it finds when it registers itself".

While this language is extremely poorly written, it can only mean one thing, namely, that LoadBalancer 270 maintains a list of look up servers 230, (registries being the same thing as look up servers) and that the list of registries is not expressly shown in 270 of Figure 2 (since some registries/look up servers are, in fact, shown in Figure 2). No other reading of paragraph [0038] would make sense. Certainly Barnett cannot

mean that LoadBalancer 270 has within it look up servers having web services. Yet this appears to be the position that the Office is taking. What is inside LoadBalancer 270 is a list of look up servers, not the look up servers themselves. Not only would it make no sense for there to be look up servers in LoadBalancer 270, but Figure 2 and the rest of the specification (particularly paragraph [0036] discussed below) make clear that the web services are available through the look up servers 230 and that the look up servers are elsewhere on the network. Particularly, Figure 2 show look up servers 230 and they are not in the LoadBlaancer 270. What is does not show is the list of look up servers in LoadBalancer 270 described in paragraph [0038]. Furthermore, paragraph [0038] is preceded by paragraph [0036], which supports Applicant's interpretation of Barnett. Paragraph [0036] describes in detail how a user's browser 205 accesses the server 200 to determine what services 240 are available to it, such as specific database 250, which Figure 2 shows as being available through one of the look up servers 230, not through the LoadBalancer 270 itself.

Accordingly, contrary to the Office's assertions, Barnett does not disclose the first element of claim 1 of "determining and describing web services software modules that are available at a corresponding, local network node".

The Second Paragraph of Claim 1

Barnett also does not meet the requirements of the second element of claim 1.

The second element of claim 1 comprises instructions "for generating messages to be transmitted to other containers via a network disclosing said web services software modules that are available at said corresponding network node". The Office asserts that this is found in paragraph [0036] of Barnett. However, as mentioned above, paragraph [0036] discloses how a first server node, web server node 200, provides information to the client browser node 205 (not to "other containers" as claimed, i.e., another server node) as to the web services that are available at other server nodes, namely, look up servers 230 (not at the "corresponding network node" as claimed).

In response to this argument as made in the previous response, the Office responded that "it is unclear what Applicant's traversal is in reference to". It appears that the Office believes that Applicant's argument did not address the specific claim language allegedly lacking from Barnett.

It is apparent from the above-noted matters that the Office did not understand Applicant's argument. Applicant's argument was and still is that:

> (1) according to the express language of claim 1, all of the elements recited in claim 1 are in a single piece of software, i.e., the container;
> (2) according to the express language of claim 1, the container is located at a single server node; and
> (3) according to the express language of claim 1, the web services software modules that are "contained" in the container reside at that same single server node.

As described above in detail, in Barnett, the software that the Office is relying on as allegedly teaching the software for generating the messages disclosing the available web services is at a different node than the web services. Therefore, Barnett cannot meet these limitations.

The Last Paragraph of Claim 1

Barnett also does not meet the recitation in last paragraph of claim 1 that the container itself is a Web service.

With respect to Applicant's previous arguments on this subject, the Office asserted that:

> A container cannot be a web service, since a web service is a method of performing a function on the web. This statement is contrary to Applicant's own claims that state a container is a web services software module.

Applicant does not understand the Office's position. The Office seems to be saying that Applicant has admitted that a container cannot be a web service by virtue of having stated that a container is a web services software module.

Is the Office considering a "web service" and a "web services software module"

to be different things? The term "web service" is simply a shortened version of "web services software module". They are one and the same thing in Applicant's application.

Dependent Claims

Some of the dependent claims are directed to very specific examples of dynamic reconfiguration of web services software modules between two (or more) servers. These include dependent claims 12-20 depending directly or indirectly from independent computer program product claim 1 and dependent claims 36-38, 40-45 depending directly or indirectly from independent method claim 25.

Each of these claims specifically recites or incorporates a specific scenario in which web services software modules are dynamically reconfigured between two or more servers. In some of the examples, web services software modules are sent over the network to other servers. In other scenarios, a server acts as a proxy for another server without actually copying, moving, or sending the web services software module over the network to another server.

All of these claims clearly patentably distinguish over Barnett, which does not disclose or discuss anything even remotely resembling proxying or moving web services software modules between two or more servers.

The Office relies on one or more of paragraphs [0035], [0037], [0039], and [0049] with respect to all of these claims. For convenience, Applicant has reproduced paragraphs [0038] through [0040] and paragraph [0049] of Barnett below.

> "[0038] More particularly, LoadBalancer 270 maintains a list of registries (lookup servers not expressly shown in 270 of FIG. 2) it finds when it registers itself, and uses all available ComputeServers available to a LoadBalancer for a job. Information about the owner of the Executable is provided by other functionality 280 and stored via a "Result" object in a Result archive 285. For example, a ResultsManager object may combine the information ("Result") with the job ID to create a unique path (URL) to the Result, sent to the user via e-mail.
>
> [0039] The LoadBalancer/ComputeServer 270 passes the job to the most

> eligible ComputeServer (not shown in 270 of FIG. 2), and identifies the ComputeServer to pass the next job to based on the total amount of time each ComputeServer of the distributed system has spent on the jobs, or based on the last time a computeServer completed a job (these are configurable options).
>
> [0040] If no LoadBalancer 270 is available, the ExecutionManager 260 returns a diagnostic message to the client applet, otherwise it invokes the LoadBalancer's run job method with the Executable as a parameter. If the LoadBalancer 270 run job is successfully passed to the LoadBalancer/ComputeServer(s) 270, the job is given a unique job I.D. that is returned to the client applet and passed along with the Executable for identification purposes.
>
> [0049] Two different types of services may be implemented by the eCommerce Software Platform architecture of this invention. The types are different from an implementation perspective. One: any service sub-classed from LocalService functions by downloading the entire object needed for execution of the service. Hence, no portion of the service is resident on any server. Such condition implies fairly limited functionality for the service because it cannot access service specific databases, or perform any remote processing."

These paragraphs merely mention "load balancing", but provide no detail whatsoever about how it is done. Thus, even if one were to improperly speculate as to the nature of the load balancing mentioned in Barnett, one can only assume that it is traditional load balancing in which the client is informed of which server has been assigned to service it and then the client communicates with that server. This is completely different than what is claimed in claims 12-20, 36-38, and 40-45.

These claims do not have anything to do with traditional load balancing. Specifically, the claims that recite specific techniques for moving the actual web services software modules about the network, i.e., claims 16-21 and 40-45, concern distributing the web services software modules themselves about the network among different servers, not distributing client requests for web services software modules.

There is no discussion whatsoever in these paragraphs of Barnett of moving web services software modules between servers. Accordingly, the subject matter of

claims 16-20 and 40-44, which concern the movement of web services software modules between servers, cannot possible be disclosed in Barnett. The only discussion anywhere in Barnett of moving web services software modules is the conventional situation where some or all of the software comprising a web service might be transmitted to the client machine. However, this has nothing to do with what is being claimed in these claims.

Furthermore, claims 12-14 and 36-38, which relate to one server acting as a proxy for another server (i.e., taking a client request, forwarding it to another server, receiving a response from the other server, and forwarding it to the requesting client), concern quite a different process than Barnett's load balancing. While proxying by one container for the web services software modules in another container in one sense achieves a similar result as traditional load balancing in that requests are distributed to a container according to load, "load balancing" *per se* is understood in the trade as being something quite different than what is claimed in these claims. Particularly, traditional load balancing deals with directing clients to send client requests to different servers in a server farm in order to balance the load of request handling at a web site, whereas the proxying claimed in these claims is a technique wherein the client machine continues to send requests to one particular server and is entirely unaware of the fact that a different server than the one it is communicating with is actually providing the Web service that it is using.

These claims recite an entirely different technique.

Furthermore, Barnett's LoadBalancer is not a container, as claimed, nor a web service software module, as claimed.

Conclusion

In view of the foregoing amendments and remarks, this application is now in condition for allowance. Applicant respectfully requests the Examiner to issue a Notice of Allowance at the earliest possible date. The Examiner is invited to contact Applicant's undersigned counsel by telephone call in order to further the prosecution of this case in any way.

Respectfully submitted,

Dated: September 26, 2007

/Theodore Naccarella/
Theodore Naccarella, Reg. No. 33,023
Synnestvedt & Lechner LLP
1101 market Street; Suite 2600
Philadelphia, PA 19107
Telephone: (215) 923-4466
Facsimile: (215) 923-2189

TXN:pmf

S:\\IBM\IBM Raleigh RSW\Patents\P25391 USA\PTO\Response-to-7 24 07 Final OX.doc